

第三次作业

No.

DATE

一、

1. 地址映射: 把逻辑地址变成内存中的物理地址
2. 动态重定位: 真正访问的内存地址是相对地址和重定位寄存器中的地址相加形成的。
3. 虚拟存储器: 具有请求调入功能和置换功能, 能从逻辑上扩大内存的一种存储器系统
4. 静态链接: 事先进行链接而以后不再拆开
5. 对换: 把内存中暂时不用的换到外存上, 把具备运行条件的进程存入内存。
6. 设备驱动程序: 是与设备无关软件和设备控制器之间通信和转换的程序
7. SPooling: 假脱机技术, 将一台物理I/O设备虚拟成多台逻辑I/O设备。
8. I/O通道: CPU和设备控制器之间, 特殊的处理机
9. 文件系统: 以便OS对自身及所有用户的数据与程序进行在线存储和访问。
10. 目标文件: 把源程序经过编译程序编译后, 但尚未经过链接
11. 文件的逻辑结构: 从用户角度来看, 文件的逻辑结构是能被存取的基本单位。
12. 有结构文件: 由一个以上的记录所构成的文件。
13. 位示图: 用二进制的位表示磁盘中一个盘块的使用情况
14. 程序接口: 用户程序取得OS服务的唯一途径。
15. 系统调用: 系统调用提供了用户程序和OS内核之间的接口
16. I/O中断: 外设触发CPU处理的硬件信号机制
17. 文件管理系统: 将文件逻辑块号转换为物理块号。
18. 文件: 文件是具有文件名的一组相关元素的集合。
19. 文件的逻辑结构: 从用户角度来看, 文件的逻辑结构是能被存取的基本单位
20. 文件的物理结构: 系统将文件存储在外存上所形成的一种存储组织形式

二、1. 重定位

2. 绝对装入方式, 可重定位装入方式, 动态运行时装入方式
3. 静态链接, 装入时动态链接, 运行时动态链接
4. 空闲分区表, 空闲分区链
5. 紧凑
6. 固定, 系统, 不固定, 用户所编写的程序
7. 对换性, 虚拟性
8. 文件, 记录, 数据项
9. 动态分区分配中所用的数据结构, 动态分区分配算法, 分区的分配与回收。

No.

DATE

10. 外部碎片 11. 分页存储管理, 分段存储管理, 段页式存储管理

12. 快表 TLB (translation look aside buffer)

13. 逻辑地址, 物理内存

14. 方便编程, 信息共享, 信息保护, 动态链接, 动态增长

15. 地址变换机构

16. 顺序文件, 索引文件, 索引顺序文件

17. 处理机, 指令类型单一, 没有自己的内存

18. 局部性, 空间局部性, 时间局部性

19. 单级索引组织方式, 多级索引组织方式, 增量式索引组织方式

20. 动态重定位 21. 与设备无关的 I/O 软件, 设备驱动程序, 中断处理程序

22. 顺序结构, 链接结构, 索引结构, 索引结构

23. CPU 寄存器, 主存储器, 辅助存储器

24. 连续分配存储管理, 离散分配存储管理

25. 存储介质的存储性能, 利用的外存分配方式

26. 顺序文件, 索引文件, 索引顺序文件

27. 连续组织方式, 链接组织方式, 索引组织方式

28. 相对路径, 绝对路径

29. 空闲盘块链, 空闲盘区链

30. 利用指向无环图实现文件共享, 利用符号链接

31. 块设备, 字符设备, 独占设备, 共享设备, 虚拟设备

32. 直接存储器访问 (DMA) 方式, I/O 通道方式

33. 单缓冲, 环形缓冲, 缓冲池 34. 寻道时间, 传输时间

三、1. 至少具有 3 层: 最高层为 CPU 寄存器, 中间层为主存储器, 最低层为辅助存储器。层次越高, 存储介质的访问速度越快, 存储容量越小。CPU 寄存器和主存是易失性存储器。

2. (1) 编写源代码 (2) 编译生成可重定位的目标文件 (3) 链接器生成可执行文件

(4) 操作系统将可执行文件装入内存, 建立 PCB 并分配资源 (5) CPU 从内存读取指令并执行

(6) 程序执行完毕

3. (1) 静态链接 (2) 装入时动态链接 (3) 运行时动态链接

异: 静态链接在编译时合并库代码, 动态链接在运行时加载, 运行时链接由程序显式控制

4. 顺序分配算法, 索引分配算法

1) 顺序分配算法中包括

(1) 首次适应算法: 从内存起始地址开始顺序搜索, 选择第一个足够大的空闲分区

(2) 循环首次适应算法: 从上次分配结束的位置开始搜索, 选择第一个足够大的空闲分区

(3) 最佳适应算法: 将空闲分区按容量从小到大排序, 选择第一个足够大的空间

(4) 最坏适应算法: 将空闲分区按容量从大到小排序, 选择第一个足够大的空间

索引分配算法包括

(1) 快速适应算法: 根据进程的长度在索引表中找到能容纳的最小空闲分区, 从链表中取第一块分配

(2) 伙伴系统: 若存在大小为 2^{i+1} 的一个空闲分区, 则把该空闲分区分为相等的两个分区, 一个分区用于分区己, 另一个分区加入大小为 2^i 的空闲分区链中

(3) 哈希算法: 构造以空闲分区大小为关键字的哈希表, 根据所需空间大小, 通过哈希函数计算得到哈希表中的位置, 从而得到相应的空闲分区链表。

5. 内存动态分区: 首次适应, 循环首次适应, 最佳适应, 最坏适应

CPU调度: 先来先服务, 短作业优先, 优先级调度, 时间片轮转, 多级反馈队列

相同点: 高效分配资源 (内存或CPU)

可能导致饥饿 / 产生碎片

不同点: 内存管理资源为物理内存的空闲分区, CPU为CPU时间片

内存的分配单位是分区, CPU是进程 / 线程

6. 页面置换: 先进先出页面置换算法, 最佳页面置换算法, 最近最久未使用页面置换算法

相同点: 高效分配资源 (内存页框或CPU时间)

不同点: 页面置换的资源类型是内存页框, CPU是CPU时间

7. 磁盘调度算法: 先来先服务, 最短寻道时间优先, SCAN算法

页面置换算法管理资源为内存页框, 磁盘调度算法为磁盘磁头 (物理存储设备)

8. 分页存储: 将逻辑地址空间和物理内存划分为固定大小的页

通过页号查页表, 获取物理页框号, 物理地址 = 物理页框号 + 页内偏移

分段存储: 按程序逻辑划分为可变长度的段

通过段号查段表, 获取段的基地址和段限长, 检查偏移量是否越界, 物理地址 = 段基地址 + 段内偏移

段页存储: 先分段再分页, 逻辑地址 = 段号 + 段内页号 + 页内偏移

先查段表获得页表基址, 然后查页表获取物理页框号, 物理地址 = 物理页框号 + 页内偏移

9. (1) 测定是否有和响应的中断信号 (2) 保护被中断进程的CPU现场环境

No.

DATE

(3) 转入相应设备的中断处理程序 (4) 处理中断 (5) 恢复CPU现场环境后退出中断

10. 用途: ① 系统文件 ② 用户文件 ③ 库文件

数据类型: ① 源文件 ② 目标文件 ③ 可执行文件

组织和管理方式: ① 普通文件 ② 目录文件 ③ 特殊文件

11. 单级目录结构, 两级目录结构, 树形目录结构, 无环图目录结构,

线性检索法, Hash方法

12. 顺序结构 { 优点: 存储效率高, 适合流式操作, 实现简单

缺点: 随机访问性能差, 插入/删除困难

索引结构 { 优点: 随机访问高效, 支持复杂查询, 动态扩展性好

缺点: 存储开销大, 写入性能较低

索引顺序结构 { 优点: 可以随机访问, 动态扩展性好

缺点: 存储开销大

13. 内存: 连续分配 { 固定分区

动态分区

离散分配 { 分页

分段

段页式

外存: 连续分配: 空闲区表法

链接组织: 空闲链表法 { 空闲盘块链

索引组织 { 空闲盘区链

成组链接法

虚拟内存管理: 页面置换

14. ① 改进文件的目录结构以及检索目录的方法, 以减少对目录的查找时间

② 选取好的文件存储结构, 以提高对文件的访问速度

③ 提高磁盘I/O速度, 以实现将文件中的数据快速地由磁盘传送到内存中, 或反向传送。